

15.11.2004

日 本 国 特 許 庁
JAPAN PATENT OFFICE

REC'D 13 JAN 2005

WIPO

PCT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 1 1 月 1 1 日
Date of Application:

出 願 番 号 特 願 2 0 0 3 - 3 8 0 6 5 6
Application Number:
[ST. 10/C] : [J P 2 0 0 3 - 3 8 0 6 5 6]

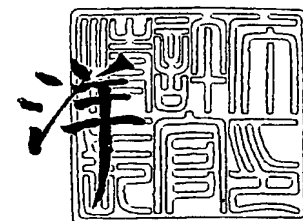
出 願 人 豊和工業株式会社
Applicant(s):

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

2 0 0 4 年 1 2 月 2 2 日

特許庁長官
Commissioner,
Japan Patent Office

小 川



【書類名】 特許願
【整理番号】 491000
【提出日】 平成15年11月11日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 15/20
【発明者】
 【住所又は居所】 愛知県一宮市大和町戸塚字毛受田 27-47 戸塚マンション B 506
 【氏名】 芹沢 一明
【発明者】
 【住所又は居所】 岐阜県岐阜市田生越町 1 丁目 1 番地
 【氏名】 田中 彰一
【発明者】
 【住所又は居所】 岐阜県岐阜市北一色 7 丁目 28 番地 16
 【氏名】 石田 勝義
【特許出願人】
 【識別番号】 000241588
 【氏名又は名称】 豊和工業株式会社
 【代表者】 野崎 東太郎
 【電話番号】 052-408-1306
【手数料の表示】
 【予納台帳番号】 001409
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1

【書類名】 特許請求の範囲**【請求項 1】**

生産システムを構成する各種構成要素について構成要素の動作シミュレーションを行なうシミュレーションプログラムを記述した要素定義ファイルを準備しておくとともに、これらの要素定義ファイルのシミュレーションプログラムに従って、関連する構成要素間でデータをやり取りして生産システムのシミュレーションを行なう工程シミュレーションシステムにおいて、生産システムの各種構成要素の配置を、工程識別名称と構成要素名称との組み合わせにより記述すると共にワークの流れに沿って記述した生産システムの要素配置データと、その要素配置データから構成要素名称を工程識別名称と組み合わせた状態で順次読み込み、読み込んだ構成要素名称に対応する要素定義ファイルから一連のシミュレーションプログラム配列を生成するシミュレーションプログラム配列作成手段と、そのシミュレーションプログラム配列で作られたシミュレーションプログラムを実行するプログラム実行手段とを備えた工程シミュレーションシステム。

【請求項 2】

要素配置データは、文字情報の追加削除等の編集が可能なソフトウェアを用いて作られる行、列データであり、列にワーク流れ方向が設定され、行に工程識別名称と構成要素名称とが記述されていることを特徴とする請求項 1 記載の工程シミュレーションシステム。

【請求項 3】

要素定義ファイルは、プログラム記述部と変数記述部とを含み、変数記述部には自己のプログラムで使用する変数の他、他の要素定義ファイルの変数を参照する要素定義ファイルにおいてはその変数記述部には、参照先変数を指定して外部参照変数を定義し、他の要素定義ファイルに変数を参照させる要素定義ファイルにおいては取出変数を定義し、要素定義ファイルを読み込んで変数記述部の全ての変数を変数配列に配列する変数配列作成手段と、前記外部参照変数と取出変数とを対応付ける変数対応手段を備えていることを特徴とする請求項 2 記載の工程シミュレーションシステム。

【請求項 4】

要素配置データには、要素定義ファイルに記述されている変数名を別の変数名に置換するための変数名置換データが記述されており、当該変数名置換データが記述されている変数名を前記別の変数名とする変数名置換手段を備えていることを特徴とする請求項 3 記載の工程シミュレーションシステム。

【請求項 5】

変数配列内に、複数の構成要素間に亘る複数の同種変数をまとめて管理するための登録エリアを必要数備えた変数テーブルを作成し、まとめて管理される同種変数を含んでいる要素定義ファイルには、該変数に応じて作成された対応する変数テーブルに、データとして変数配列内における該変数の変数位置を登録する登録手段を含んでいることを特徴とする請求項 3 または 4 記載の工程シミュレーションシステム。

【請求項 6】

シミュレーションプログラム記述部のシミュレーションプログラムは、ラダー言語方式の命令群で記述されていることを特徴とする請求項 1～5 記載の工程シミュレーションシステム。

【書類名】明細書

【発明の名称】工程シミュレーションシステム

【技術分野】

【0001】

この発明は、複数の加工ユニット、クランプユニット、搬送ユニット等が相互連繋により動作する、例えば、トランスファラインのような生産システムの工程シミュレーションシステムに関するものである。

【背景技術】

【0002】

従来、このような工程シミュレーションシステムとして、特許文献1には、生産システムの各構成要素のシミュレーションモデルを表形式で表現し、内部記憶部に展開して、関連するモデルの間で関連フレームリストを参照しつつデータのやり取りを行いながらシミュレーションを実行するものが開示され、オペレータ、生産セル、搬送系に関するモデルが例示されている。

また、特許文献2には、工作機械、段取りステーション、搬送装置、加工ワークなどに関する変数データを入力データファイルに入力し、その変数データに基づいて工程シミュレーションを行ない、シミュレート結果を出力するものが記載されている。

特許文献3には、LSI製造工程の装置夫々について、シミュレーションモデル及びパラメータが別個に設定できるものが開示されている。

【特許文献1】特開平4-64164号公報

【特許文献2】特開昭61-61752号公報

【特許文献3】特開平10-335193号公報

【発明の開示】

【発明が解決しようとする課題】

【0003】

本発明は、これらの従来技術に鑑み、シミュレーションプログラム生成が容易なシミュレーションシステムを提供しようとするものである。また、工程の増加、削除等、要素配置データの編集が容易な上記システムを提供することを目的とする。

【課題を解決するための手段】

【0004】

この発明の工程シミュレーションシステムは、生産システムを構成する各種構成要素について構成要素の動作シミュレーションを行なうシミュレーションプログラムを記述した要素定義ファイルを準備しておくとともに、これらの要素定義ファイルのシミュレーションプログラムに従って、関連する構成要素間でデータをやり取りして生産システムのシミュレーションを行なう工程シミュレーションシステムにおいて、生産システムの各種構成要素の配置を、工程識別名称と構成要素名称との組み合わせにより記述すると共にワークの流れに沿って記述した生産システムの要素配置データと、その要素配置データから構成要素名称を工程識別名称と組み合わせた状態で順次読み込み、読み込んだ構成要素名称に対応する要素定義ファイルから一連のシミュレーションプログラム配列を生成するシミュレーションプログラム配列作成手段と、そのシミュレーションプログラム配列で作られたシミュレーションプログラムを実行するプログラム実行手段とを備えていることを特徴とする。

要素配置データは、文字情報の追加削除等の編集が可能なソフトウェアを用いて作られる行、列データであり、列にワーク流れ方向が設定され、行に工程識別名称と構成要素名称とが記述されている。

要素定義ファイルは、プログラム記述部と変数記述部とを含み、変数記述部には自己のプログラムで使用する変数の他、他の要素定義ファイルの変数を参照する要素定義ファイルにおいてはその変数記述部には、参照先変数を指定して外部参照変数を定義し、他の要素定義ファイルに変数を参照させる要素定義ファイルにおいては取出変数を定義し、要素定義ファイルを読み込んで変数記述部の全ての変数を変数配列に配列する変数配列作成手

段と、前記外部参照変数と取出変数とを対応付ける変数対応手段を備えている。

【0005】

要素配置データには、要素定義ファイルに記述されている変数名を別の変数名に置換するための変数名置換データが記述されており、当該変数名置換データが記述されている変数名を前記別の変数名とする変数名置換手段を備えている。

変数配列内に、複数の構成要素間に亘る複数の同種変数をまとめて管理するための登録エリアを必要数備えた変数テーブルを作成し、まとめて管理される同種変数を含んでいる要素定義ファイルには、該変数に応じて作成された対応する変数テーブルに、データとして変数配列内における該変数の変数位置を登録する登録手段を含んでいる。

シミュレーションプログラム記述部のシミュレーションプログラムは、ストアードプログラムサイクリック方式言語、一般的にはラダー言語方式、の命令群で記述されている。

【発明の効果】

【0006】

本願装置では、生産システムの要素配置データを工程識別名称と構成要素名称との組み合わせで記述し、その要素配置データに従って読み込んだ構成要素名称から対応する要素定義ファイルを読み込んでそこに記述されているシミュレーションプログラムをシミュレーションプログラム配列に格納してシミュレーションプログラムを作り、これを実行するようにしたので、要素配置データ上で異なる配置位置にある同一の構成要素は、工程識別名称との組み合わせにより別の構成要素として識別され、同一の構成要素の要素定義データについては、1種類のみ記述しておけば足り、シミュレーションプログラムを生成するに当たっての要素定義データの作成作業が容易となる。また、要素配置データに基づいて、シミュレーションプログラム配列が作成されるので、別のシミュレーションを行なうために要素配置データが変更されたときでも、その変更された配置に対するシミュレーションプログラムを容易に生成できる。

また、文字情報を入力し、追加削除等の編集が可能なソフトウェアを用いて、要素配置データを作成したので、その要素配置データに新たに工程識別名称と構成要素名称の組み合わせを追加したり、削除することがソフトウェアの編集機能を利用して容易に行なうことができ、工程の増加、削除に関して要素配置データを極めて容易に作成できる。

まとめて管理すると都合のよい変数、例えば、機械の故障状態といった変数について、要素定義ファイルに予めその旨を記述しておき、それらの変数を予め準備した変数テーブルでまとめて管理するようにしたので、機械の故障状態を探して作業員が修理に向かう、などのシミュレートをするときに、そのまとまった変数テーブルに対する変数状態の判別を行なうことにより機械の故障状態を探すといったシミュレーションを短い実行時間で行ない得る。

要素定義ファイル内に定義されている変数名称を、外部から別の変数名称に置換できるので、例えば要素定義ファイルにおいて外部参照変数の参照先名称を一般的な変数名称で記述してこれを要素配置データ上の具体的な参照先名称に置換することで、1つの要素定義ファイルを用いて参照先として種々の参照先を指定できる。また、同一加工工程において、同種の加工ユニットが複数配置されるときには、加工ユニットを示す一般的変数を加工ユニットを区別する別の変数名に置換することにより、これらのユニットを、同じ加工ユニットの要素定義ファイルを用いて区別することができる。

また、シミュレーションプログラムをラダー言語方式で記述したので、複雑な分岐命令を使うことなく、構成要素の動作シミュレーションを記述できる。

【実施例】

【0007】

図1において、本願システム全体構成を示す。シミュレーションシステム1は、シミュレーション目的のために作られたコンピュータプログラムをパーソナルコンピュータPC上で動作させて実現されるものであり、図7に示すフローチャートに従うシステムプログラムにより構成されるシミュレーション手段2により、要素配置データ3と要素定義ファイル4を参照しつつ、後述する要素配列5、変数配列6、外部参照変数対応配列7、初期

処理プログラム配列8、シミュレーションプログラム配列9を作成する。また、プログラム実行手段10は、作成された初期処理プログラム配列8、シミュレーションプログラム配列9に配列された初期処理プログラム、シミュレーションプログラムを実行して、初期処理と動作シミュレーションを行なうものである。パーソナルコンピュータPCには、ソフトウェアを起動、停止させるためのマウスやキーボードなどの入力手段11や、シミュレーション結果を表示するCRTなどの表示装置12が接続されている。

【0008】

以下、生産システムとして、前ストッカのワークをコンベアで接続台に送り込み、その接続台に送り込まれたワークを、リフトアンドキャリー方式の搬送装置で加工ステーション(140T・1ST)と加工ステーション(140T・2ST)で加工して、排出側の接続台に排出し、その排出側の接続台からコンベアで搬出して後ストッカにストックする、というシステムで説明する。このシステムには、機械保全のための人員が2名(保全メンバA、B)居るものとする。

【0009】

このような生産システムのシミュレーションを行なうとき、本願シミュレーションシステム1では、先ず、シミュレートしようとする生産システムを構成する構成要素(装置)に関する要素配置データ3を準備する。要素配置データ3は、基本的には工場内の装置配置図である。この要素配置データ3は、文字情報を追加削除、置換などの編集することのできるソフトウェア、例えば、汎用の表計算ソフト(MICROSOFT社のEXCEL(登録商標))で作成される表形式のデータシートである。表形式のデータシートは、テキストエディタのような編集ソフトで作ってもよい。このデータシートは、図2に示すように、基本的には、列(表の縦方向:行並び方向)にワーク流れ方向が設定され、行(表の横方向:列並び方向)には、工程識別名称と構成要素名称とが、工程識別名称で区別される工程ごとに1行ずつ記述されている。具体的には、各行のB列セルにモジュール名が、C列セルには工程名が、D列セル以後の任意なセルには、構成要素名称が記述されている。工程識別名称は、同一行のモジュール名と工程名から成り、工程名が省略されている場合もある。1つの要素配置データ3内で、工程識別名称は全て互いに異なる名称となるように記述されている。

【0010】

構成要素は、同一仕様のものについては同じ名称で記述されている。ここでは、2つの加工ユニットがすべて同一の"ユニット1"であり、2つのジグが全て同一の"ジグ1"であり、さらに2つのコンベア、接続台、保全メンバも夫々同一の仕様であるものとして、各同一名称で記述している。実際の工場には配置されることのない構成要素が、この要素配置データ3に記述されている。構成要素"総合1"は、モジュール140T内で、シミュレーション実行時にまとめて管理するのが便利に変数情報を集めておくための要素であり、後述のように、2つのジグ1のアンクランプに関する変数をまとめて管理する"アンクランプT"テーブルや図示しないが、各ユニット1の"原位置"データをまとめて管理する変数テーブルを持ち、LF搬送に対して、そのアンクランプ情報を基にして、全てのジグがアンクランプし、かつ、全てのユニットが原位置にある時、搬送可能という情報を流すというシミュレーションを行なう。また、構成要素"保全指令1"は、生産システム内でのシミュレーション実行時にまとめて管理するのが便利に変数情報を集めておくための要素であり、具体的には後述するが、例えば、全ての機械の稼動状況や保全メンバの作業状況などのデータをまとめておく。

【0011】

これらの構成要素名称の夫々と対応して、図3～5に示す要素定義ファイル4、4a～4jが夫々予め準備されている。各要素定義ファイル4の一般的構成としては、図3に示すように、先頭に要素名称記述部4A、続いて変数記述部4B、プログラム記述部4Cとから成る。変数記述部4Bには、自己のプログラム内で使用する内部変数が初期値と共に記述される他(例えば、ユニット1(加工ユニットである)では、加工時の前進速度や後退速度など)、他の要素定義ファイルの変数を参照する要素定義ファイルにおいてはその

変数記述部 4 B には、参照先変数を指定して外部参照変数を定義し、他の要素定義ファイルに変数を参照させる要素定義ファイルにおいては取出変数を初期値と共に定義するようになっている。

取出変数、外部参照変数を具体的に説明する。外部参照変数は、次のルールで記述されている。

自ファイルでの変数名 (外部参照変数) 参照先変数名 (又は変数テーブル名) 参照先工程名 モジュール名

【0012】

参照先工程名、モジュール名は、その外部参照変数が定義される要素定義ファイルの構成要素に対して、相対的な位置関係を示す語を用いて記述されてもよいし、特定のモジュール名、工程名を記述してもよい。相対的位置関係を示す語、とは、“前工程” “自工程” “次工程”、あるいは“前モジュール”、“自モジュール”、“次モジュール”といった記述である。“前工程”は、自要素に対して要素配列データ 3 上で同じ列の上側にある構成要素のモジュール名と工程名を示し、“自工程”は、自要素と同一行の構成要素のモジュール名と工程名を示し、“後工程”は、自要素に対して同じ列の下側にある構成要素のモジュール名と工程名を示す。前、自、次モジュール、についても、自モジュールに関する相対的位置を示している。参照先変数名 (または参照先変数テーブル名) は、指定されたモジュール、工程内で他の変数名と重複しないようにしてある。

【0013】

例えば、ユニット 1 とジグ 1 とでは、同じ工程に含まれるジグ 1 のクランプ完了でユニット 1 が加工を開始し、加工が完了するとその加工完了信号でジグ 1 がアンクランプする、というシーケンス動作が行なわれるが、この情報のやり取りを行なうために、図 5 のユニット 1 の要素定義ファイル 4 g においては、“自工程加工準備完了” (外部参照変数) として“自工程”の“加工準備完了” (参照先変数名) を参照する、と定義されている。これに対して、ジグ 1 では、取出変数として“加工準備完了” (ワーククランプの完了の意味である) が定義されている。また、ジグ 1 では外部参照変数としての“ユニット加工完了”を“自工程”の“加工完了”を参照すると定義し、ユニット 1 では、“加工完了”を取出変数としている。従って、これらのジグ 1 とユニット 1 とが同じ工程に配置された状態では、ユニット 1 と同じ行 (つまり、自工程) に配列されたジグ 1 との間で取出変数と外部参照変数が対応することになる。他にも同様の取出、外部参照の関係が、図 4 においては、接続台とコンベア、接続台と LF 搬送との間で例示されている。

【0014】

参照先変数が変数テーブルである例を説明する。図 5 のユニット 1 では、

“ 保全指令・機械状態 機械状態 T NONE 保全指令 ”

と外部参照変数が定義されている。“NONE”は、工程名が無いことを示している。要するに、変数“保全指令・機械状態”は、モジュール名“保全指令”の“機械状態 T”という変数テーブルに対応する、ということである。その他の構成要素でも“機械状態”に関する外部参照変数が定義してあり、ジグ 1 の要素定義ファイル 4 f や LF 搬送の要素定義ファイル 4 e にも同様の記述が例示されている。これに対応するように、図 5 における“保全指令 1”の要素定義シート 4 i には、取出変数として、

[1 1] 機械状態 T

と記述されている。[] 内は装置数である。この記述があるときには、後述の変数配列 6 作成時に、要素定義ファイル“保全指令 1”を読み込んだときの配列エリア中に、装置数 (ここでは 1 1) + 1 個のデータ登録エリアを展開して、図 12 (A) に示すように、“機械状態 T [0] ~ 機械状態 [1 1]”までのテーブル変数を持つ“機械状態 T”という変数テーブル 20 を作るようになっていく。保全指令 1 と総合 1 を除く各要素 (保全指令 1 と総合 1 は、架空要素であるから変数“機械状態”はない) の“保全指令・機械状態”という外部参照変数は、この取出変数としてのテーブル変数“機械状態 T [1] ~ [1 1]”に順に対応付けられるようにしてある。同様の変数テーブル 20 として、図 9、10 に示す変数配列 6 には、保全指令 1 で作られる“作業員状態 T”テーブルや、総合 1 で作られる

” アンクランプ T” テーブルが例示されている。

【0015】

要素配置データ 3 上で配置した結果、外部参照変数の参照先を、単純に自工程の前、次工程との関連で記述できないような場合がある。たとえば、図 2 における接続台は、前後の搬送要素を繋ぐものであり、前後の搬送要素を外部参照変数の参照先としている。図 2 の要素配置データ 3 では、ワーク流れ方向の上流側のものについては、前工程は” 140 T・コンベア” であるが、後工程は” 140 T・1ST” であって、前後の参照先が整合しない。また、後ろの接続台では、前工程は” 140 T・2ST” となり、後工程は” 140 T・搬出コンベア” であって、これも参照先が整合していない。そこで、図 4 に示すように、接続台の要素定義ファイル 4 c では、外部参照変数の参照先工程名として、” 後搬送”、” 前搬送” といった一般的な変数名称で記述し、要素配置データ 3 に該一般的な変数名称を別の変数名称に置換する変数名置換データを記述して、シミュレーション手段 2 の変数名置換手段 2 a により変数名を置換するようにしている。要素配置データ 3 では、上流側の接続台では、前搬送が” コンベア” であり、後搬送が” 搬送” であり、一方、下流側のものでは、前搬送は” 搬送” であり、後搬送は、” 搬出コンベア” であるから、各接続台に関しては、前搬送、後搬送が違っているの、本来は、別々のコンベアとして要素定義ファイルを作成しなければならないが、このように変数名を置換することで、1 つの要素定義ファイルで前、後搬送工程が違っている場合にも対応できるようにしてある。

【0016】

変数名置換データは、構成要素名の設定されたセルの次セルに記述され、その記述のルールは、

置換後変数名 = 置換前変数名

であり、複数の変数を置換したい時には、この記述が同一行に並べて記述される。

このような変数置換は、外部参照変数の参照先工程名に限らず、各要素定義ファイルに記述されている変数名の全てについて置換できる。たとえば、図 16 に示すように 140 T・1ST の 2 つのユニットは、単純に” モジュール名 + 工程名” との組み合わせでは、夫々を区別できないので、この変数名置換手段 2 a で、要素名を替えてやるようにすればよい。図 16 では、上側のユニット 1 を L ユニットに、下側のユニット 1 を R ユニットと置換し、各ユニット原位置も区別するようにしている。

【0017】

プログラム記述部 4 C は初期処理プログラム記述部 4 C 1 とシミュレーションプログラム記述部 4 C 2 とを有する。初期処理プログラム記述部 4 C 1 では、前記した変数テーブル 20 に、該当する変数の変数位置を格納するためのデータ格納命令 (SET TBN0) や、シミュレーションプログラムを実行するための変数へのデータ設定などが行なわれるプログラムが記述されている。

シミュレーションプログラム記述部 4 C 2 には、夫々の構成要素の動作をシミュレートするプログラムが記述されている。前ストッカでは、後工程となるコンベアからワーク要求信号があつて、ストッカ上にワークがあるときには前ストッカ上のワークをコンベアに受け渡すといった動作、コンベアでは、ワークがない時に前工程要素にワーク要求信号を出したり、後工程要素からワーク要求があつて前工程要素からワーク受渡しされているときには、前進動作してワークを後工程要素に受け渡すといった動作、接続台は、搬送要素と搬送要素を繋ぐものであつて、ワークの有無を判断して前搬送要素にワーク要求信号を出し、あるいは、次搬送要素にワーク準備完了信号を出すといった動作、LF 搬送では、前搬送要素からワーク準備完了信号を受け取り、しかも、自モジュール内の全てのジグ 1 がアンクランプでユニット 1 も原位置である場合には、リフトアンドキャリー動作を行ない、ワークを順次搬送するといった動作、ジグ 1 では、LF 搬送が完了したことを確認してワークをクランプし、ユニット 1 による加工完了を待ってアンクランプするといった動作、ユニット 1 では、ジグ 1 によるクランプを確認して原位置から前進し、加工し、後退して原位置に戻るといった動作、後ストッカでは、ストッカ上にワークが無ければ、前工

程要素にワーク要求するといった動作、保全メンバでは、保全を要する機械に対して、予め設定してある保全作業時間が経過したかどうかを判断して、保全作業実行中か、空いているのかなどをシミュレートするプログラムが夫々記述されている。

【0018】

これらのプログラムは、ストアードプログラムサイクリック処理方式、いわゆるラダー言語方式の命令群で記述されている。例えば、ユニット1について説明すれば、図6の左側に実際のシミュレーションプログラムが示され、右側にそのシミュレーションプログラムをラダー表示したものを示している。

”MOTION”命令は、動作のシミュレート命令であり、命令の後に、前進動作、後退動作の別が指示され、この命令が実行される度に、動作対象の位置、速度などの変数値が刻々変化していく。”TMR”はタイマのシミュレート命令であり、”前進端タイマ設定”という変数に設定された時間が経過すると接点を閉じるというものである。”MOV”命令は、前の変数値を、後の変数にコピーするもの、”OUT”は、後ろの変数をON(”1”)とするものである。また、”LD”命令は、回路ブロックの始まりであり、”LDF”は判別を伴った回路ブロックの始まりである。”AND”命令は、論理積、ANDNOTは否定の論理積、ANDFは比較を伴った論理積、”SUB”は先頭の変数から真中の変数値を差し引いて3番目の変数に入れるもの、”SET”命令は、変数に”1”をセットするものである。

【0019】

プログラムの命令群C1～C8は、ラダー図のR1～R8に対応しており、両者を対照することで上記命令の意味が理解されることと思われるが、詳細に説明すれば、命令群C1では、自工程加工準備完了(ジグがクランプ)でユニットが原位置にあり、機械が故障中で無く、加工完了していないときに前進指令を出し、命令群C2では、前進指令で前進動作させ、命令群C3では、前進動作の結果、前進端に至ったら前進端タイマをONとし、命令群C4では、前進端タイマが設定された時間経過したら加工完了とし、命令群C5、C6では、加工完了してユニットが原位置になれば後退指令を出して後退動作させるというものである。

【0020】

また、命令群C7以下は故障発生に関するシミュレーションプログラムであり、”故障発生時間count”という変数が”0”以上で、故障発生実行という変数が”0”以上(これは故障を発生させる、という意味である)であることを判断して、両方の条件が整えば、”故障発生時間count”に設定されている時間データからシミュレーションプログラムを実行させる時の1スキャン分の時間(スキャンピッチ)を差し引いて”故障発生時間count”へ格納するという意味であり、”故障発生時間count”には、各機械毎に予め定めてある故障発生までの時間(故障発生時間)を初期処理プログラムでコピーしてあるので(図5のユニット1の定義ファイル4gにおける初期処理プログラム(MOV)を参照)、ここではシミュレーション開始からの経過時間を故障発生時間から減じるという計算をしている。次の命令群C8では、命令群C7での演算結果が、”0”以下となった、つまり、故障発生するだけの時間が経過したことを判断し、そのときに機械状態が”ゼロ”(故障していない)であれば、機械状態という変数を”1”(故障)とするというシミュレーションを行なっている。命令としてはこの他にもいくつか準備され、各構成要素の動作シミュレーションを行なうのに必要な命令群を構成している。別の命令としては、変数テーブル20に対して一定の条件で検索を行なう命令や、回転動作をシミュレートするための命令などもある。このようなラダー記述方式では、ある命令群で記述した条件が整わない場合は、次の命令群を処理する、というようにシーケンシャルに処理が行なわれるので、いくつかの条件に応じていくつかの処理を行なわせるような場合、複雑なIF文で、プログラム処理の飛び先をいくつか指定するような形式の条件式に比べて、簡単な記述となって扱いやすい。

【0021】

次に図7に示すフローチャートに従って、本願システムの動作を各配列作成と共に説明

する。まず、ステップ S 1（要素配置データ読み込み手段）で、準備された要素配置データ 3 を 1 行読み込む。ステップ S 2 で読み込んだ 1 行のモジュール名、工程名、要素名を順に図 8 に示す要素配列 5 の一行に並べていく。モジュール名の前には要素配列 5 内の位置を示すアドレスが記録され、その要素配列 5 内の位置に示されるアドレスは、要素配置データ 3 を 1 行読むと 1 ずつ増えていく。また、要素名の後ろには、後述の変数配列 6 における各要素に対応した変数の格納位置が、先頭変数位置と変数個数で格納されるエリアが設けられる。このとき、要素名に関する変数名置換データがあれば、置換後の変数名で要素名を置換する。

【0022】

次に構成要素名から対応する要素定義ファイル 4 a ~ 4 j を探し出して読み込む（ステップ S 3：要素定義ファイル読み込み手段）。ステップ S 4（変数配列作成手段）で定義ファイル 4 の変数記述部 4 B に記述されている全ての変数名と値とを、変数の属性（内部変数、取出変数、外部参照変数）と共に読み出し、図 9, 10 に示すように変数配列 6 に並べる。属性の前側に変数配列 6 内での各変数の位置を示すアドレスが順に付されていく。ここでも、変数名に関する変数名置換データがあるときには、指定された変数名が読み込んだ変数名にあるか検索して、もし置換すべき変数名があれば置換後の変数名を変数配列 6 に記録する。外部参照変数は、変数名の後ろには、参照先変数名と参照先工程・モジュール名が記述されていて、具体的な数値が入っていないので、このステップ S 4 の時点では属性と変数名のみを読み込み、値は空白である（図 11（A）, 12（A））。変数テーブル 20 も同様に読み込まれる。前述のように変数テーブル 20 では、宣言された登録エリア + 1 の登録エリアを持つ変数テーブル 20 に展開されて変数配列 6 内に登録される。

【0023】

また、外部参照変数が記述されている要素定義ファイルでは、外部参照変数名、参照先変数名、参照先工程、モジュール名が一行ずつ読み出され、図 13 に示す外部参照変数対応配列 7 を生成する。この配列を生成するときも、変数名置換データに、参照先要素名の置換変数名があれば、置換後の参照先要素として配列に登録する。例えば、140 T・接続台・接続台では、要素定義ファイル 4 c に記述の外部参照変数において、“前搬送”が“コンベア”に置換され、“後搬送”が“搬送”と置換されて登録される。

次にステップ S 6 で、要素定義ファイル 4 に初期処理プログラムがあれば、その初期処理プログラムが読み出されて図 14 に示すように初期処理プログラム配列 8 に登録される。ステップ S 7（シミュレーションプログラム配列作成手段）では、要素定義ファイル 4 のシミュレーションプログラムが読み出されて図 14 に示すようにシミュレーションプログラム配列 9 に登録される。これらのプログラムの読み込みの際、プログラム中で使用する変数は、前記変数配列 6 に配列された当該変数の変数位置（アドレス）と対応付けられる。以上の処理が、要素配列 5 データを 1 行読んで、1 つの要素定義データを読み込むたびに行なわれ、この作業が、要素配列 5 データの末行まで行なわれるとステップ S 8 で全要素配置についての読み込みが完了する。

【0024】

この時点では、外部参照変数と参照先変数との対応が取れていないので（図 11（A））、ステップ S 9（変数対応手段）でその対応処理が前記外部参照変数対応配列 7 を参照しながら行なわれ、変数配列 6 において空白となっている外部参照変数の値／位置の欄に、対応する取出変数の変数配列 6 内の変数位置（アドレス）が書き込まれる。例えば、図 11（A）において 140 T・1 S T・ユニット 1 の外部参照変数“自工程加工準備完了”には、同じ工程のジグ 1 の取出変数“加工準備完了”が対応しているので、外部参照変数“自工程加工準備完了”には、アドレス R 5 が図 11（B）のように登録される。

【0025】

取出変数として変数テーブル 20 が登録されている場合には、その変数テーブル 20 に対応する外部参照変数の値／位置欄には、変数テーブル 20 の変数配列 6 内での位置が順に登録されていく。図 12（A）に示すように、前述の“機械状態 T”という変数テー

ル 20 の各テーブル変数には、各機械の外部参照変数” 保全指令・機械状態” が対応している。変数配列作成当初には空白であった各機械の” 保全指令・機械状態” の値／位置欄、例えば” 保全指令・機械状態” という変数を持つ要素配置データ 3 で 6 つ目の要素 (140T・1ST・ユニット 1) の変数” 保全指令・機械状態” には、変数テーブル” 作業状態 T” の 6 番目の取出変数の変数位置 (ここでは T0+6) が登録される (図 12 (B))。

【0026】

次に、ステップ S10 で初期処理プログラムが実行される。初期処理プログラム配列 8 に並んだ初期処理プログラムが一命令ずつ、プログラム実行手段 10 により実行される。初期処理プログラムとして、” SET TBN0” 命令が実行されると、そこで指定された変数の変数配列 6 内位置データを、指定された外部参照変数の示すテーブル変数の値／位置欄に書き込み、同時に、そのテーブル変数の先頭テーブル変数” 作業状態 T[0]” のデータとしての登録個数を” 1” 加算する、という作業が行なわれる。上述の 140T・1ST・ユニット 1 では、

SET TBN0 ” 機械状態” データ位置 保全指令・機械状態

と記述され、変数” 機械状態” の変数配列 6 内位置データ (Q6) を” 保全指令・機械状態” の示すアドレス” T0+6” の変数” 機械状態[6]” の値欄に書き込むということが行なわれる (図 12 (C))。同様の処理が、「アंकランプ T」という変数テーブル 20 と 2 つのジグ 1 の変数” アंकランプ” との間で行なわれている点が例示されている (図 9)。” SET TBN0” 命令は、変数テーブル 20 に対応する変数位置とテーブルへの登録数を登録する登録手段を構成する。

【0027】

こうして、図 9、図 10 に示すように、変数配列 6 に、シミュレーションで用いる全ての内部変数、外部参照変数、取出変数が並べられ、外部参照変数と取出変数が対応付けられ、かつ、まとめて管理すると好適な変数とその変数をまとめて管理する変数テーブル 20 とが対応付けられたので、全ての変数は、この変数配列 6 を参照することで値を知ることができる。

ついで、入力手段からシミュレーションの指示があると (ステップ S11)、ステップ S12 では、シミュレーション配列に要素配列 5 に従って並べられたシミュレーションプログラムが、プログラム実行手段 10 によって先頭から順番に記述された命令に従って実行される。シミュレーションプログラムは、各命令の記述が文字情報であるので、その命令を読む都度、プログラム実行手段 10 が文字情報による命令記述を解釈して実行してもよいが、予め各命令をそれらの各命令に対応して設定された数値に置き換えておき、その命令に対応する数値を読み出して、その数値に対応した命令処理をプログラム実行手段 10 が実行するようにすると、実行速度が上がって好ましい。シミュレーションプログラムが実行されると、変数配列 6 の対応する変数の値が刻々と変化する。

【0028】

シミュレーションプログラムは先頭から末尾まで一定のスキャンピッチ (シミュレーションする上での経過時間: 例えば 0.1 秒間隔) で繰り返し実行され、実行に当たって、変数配列 6 の各変数が参照される。このスキャンピッチは任意に設定でき、たとえば、実際の機械の動作時間 (実時間) でシミュレーションするように設定することもできるし、逆に短くして早送りのようにシミュレーションすることも可能である。外部参照変数では、参照先変数が、その外部参照変数の示すアドレス (変数配列内位置) を頼りに読み出される。シミュレーションする上で所定の時間経過した時点で (あるいは 1 スキャン終わるごとに)、実行を停止できる。ステップ S13 により、構成要素を特定してシミュレーション結果表示が指示されると、要素配列 5 から変数配列 6 の変数格納範囲が読み出され、コンピュータに接続された出力装置 (CRT) 12 上に、工程識別名称、構成要素名と共に変数名とその値が表示され、これにより、シミュレーション結果を確認できる。構成要素を表示画面上でアニメーション表示するようにし、変数値の変化をアニメーションで動作表示させると、なお判りやすい。

【0029】

シミュレーションプログラムの実行時、保全指令1では、“機械状態T”テーブルを用いて全ての機械状態（異常なしか、故障中か）を判断し、“作業員状態T”テーブルにより空いている保全メンバを検索して、空いている保全メンバを要修理機械の修理に向かわせる、といったシミュレーションが行なわれる。

【0030】

このようなシミュレーションにおいて、変数テーブル20として“機械状態T”テーブルを持たなくても、変数配列6を先頭から末尾まで順に検索すれば、全ての“機械状態”データを知ることができるが、本願では、構成要素ごとの機械状態データについて、その変数位置（変数配列6内でのアドレス）をまとめて“機械状態T”テーブルに集約しているため、その限られた記憶領域のデータについて参照することで短時間で全ての“機械状態”データを知ることができ、シミュレーションプログラム実行時間を短くできる。

【0031】

こうしてある要素配置データ3に関するシミュレーションを行なうことができるが、更に140T・2STの後に140・2STと同じ工程を増やしてシミュレーションしたい場合には、要素配置データ3に、表計算ソフトの編集機能を利用して140T・2STの1行をコピーして140T・2STの下に1行追加してこれを140T・3STとし、この加工工程（ジグ1、ユニット1）の追加により変更しなければならない関連するデータ（例えば、“アंकランプT”テーブルのデータ登録エリア数、“機械状態T”テーブルのデータ登録エリア数など）を修正して、図7に示すプログラムを実行してやれば、追加したジグ1とユニット1間での変数間対応については、要素配置データ3上での配置により、ジグ1、ユニット1の要素定義データにおいて記述されている相対的な参照先指示により一義的に定まるから、要素定義データに変更がない限り、新たに追加した工程を含んだシミュレーションプログラム配列9、変数配列6、その他の配列が作成されることになり、そのようにして作られたシミュレーションプログラム配列9で並べられた一連のシミュレーションプログラムを実行すれば、別の工程編成による工程シミュレーションが容易に実行できる。

【図面の簡単な説明】

【0032】

- 【図1】本願シミュレーションシステムの全体ブロック図である。
- 【図2】要素配置データを示す図である。
- 【図3】要素定義ファイルの一般的形態を示す図である。
- 【図4】要素定義ファイルの具体例である。
- 【図5】要素定義ファイルの具体例である。
- 【図6】シミュレーションプログラムの具体的記述を説明する図である。
- 【図7】本願シミュレーションシステムのフローチャートである。
- 【図8】要素配列を示す図である。
- 【図9】変数配列を示す図である。
- 【図10】変数配列を示す図であり、図9の続きである。
- 【図11】外部参照変数と取出変数の対応を説明する図である。
- 【図12】変数テーブルと外部参照変数との対応を説明する図である。
- 【図13】外部参照変数対応配列を示す図である。
- 【図14】初期処理プログラム配列を示す図である。
- 【図15】シミュレーションプログラム配列を示す図である。
- 【図16】変数名置換データを適用した他の例である。

【符号の説明】

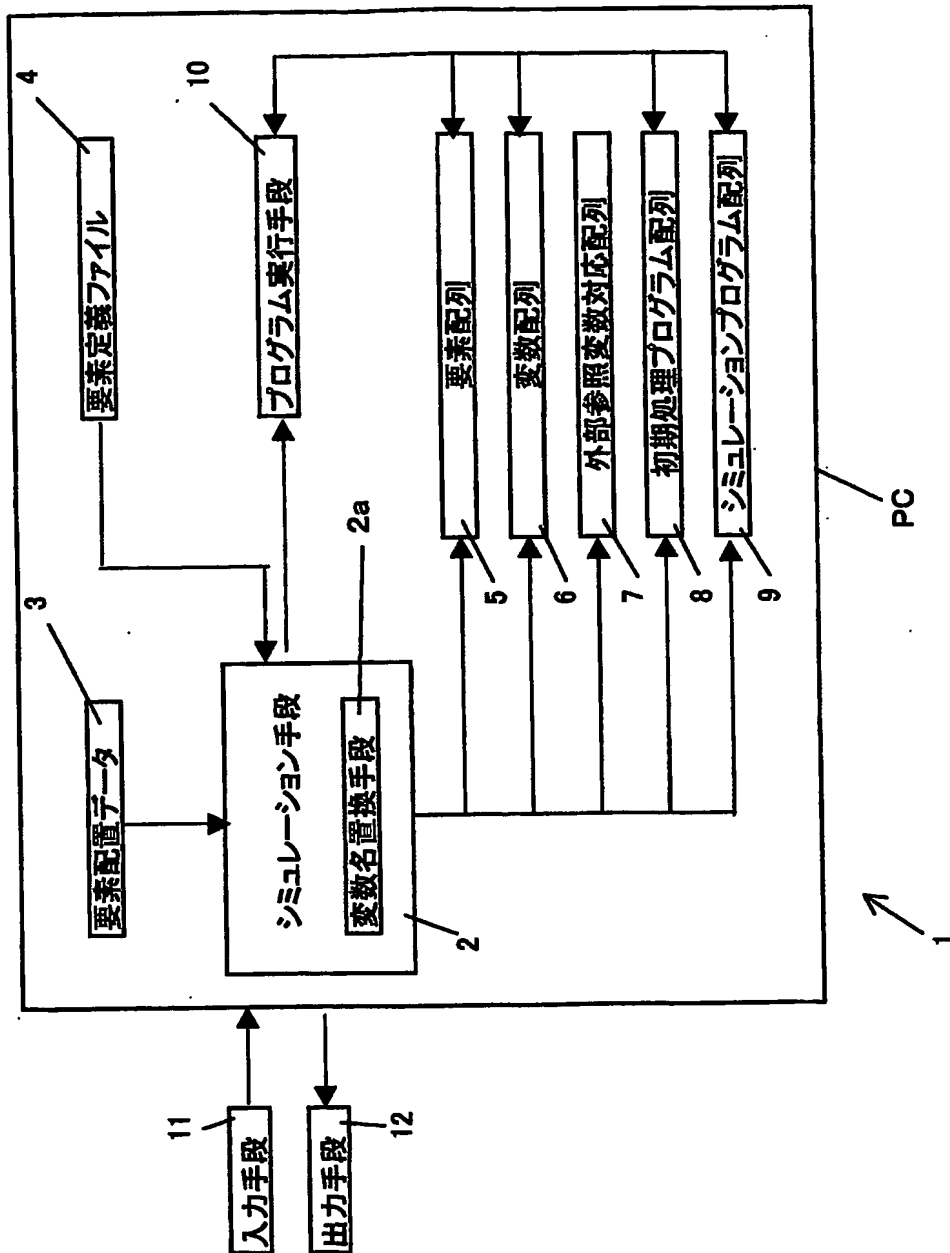
【0033】

- 1 シミュレーションシステム
- 2 a 変数名置換手段
- 3 要素配置データ

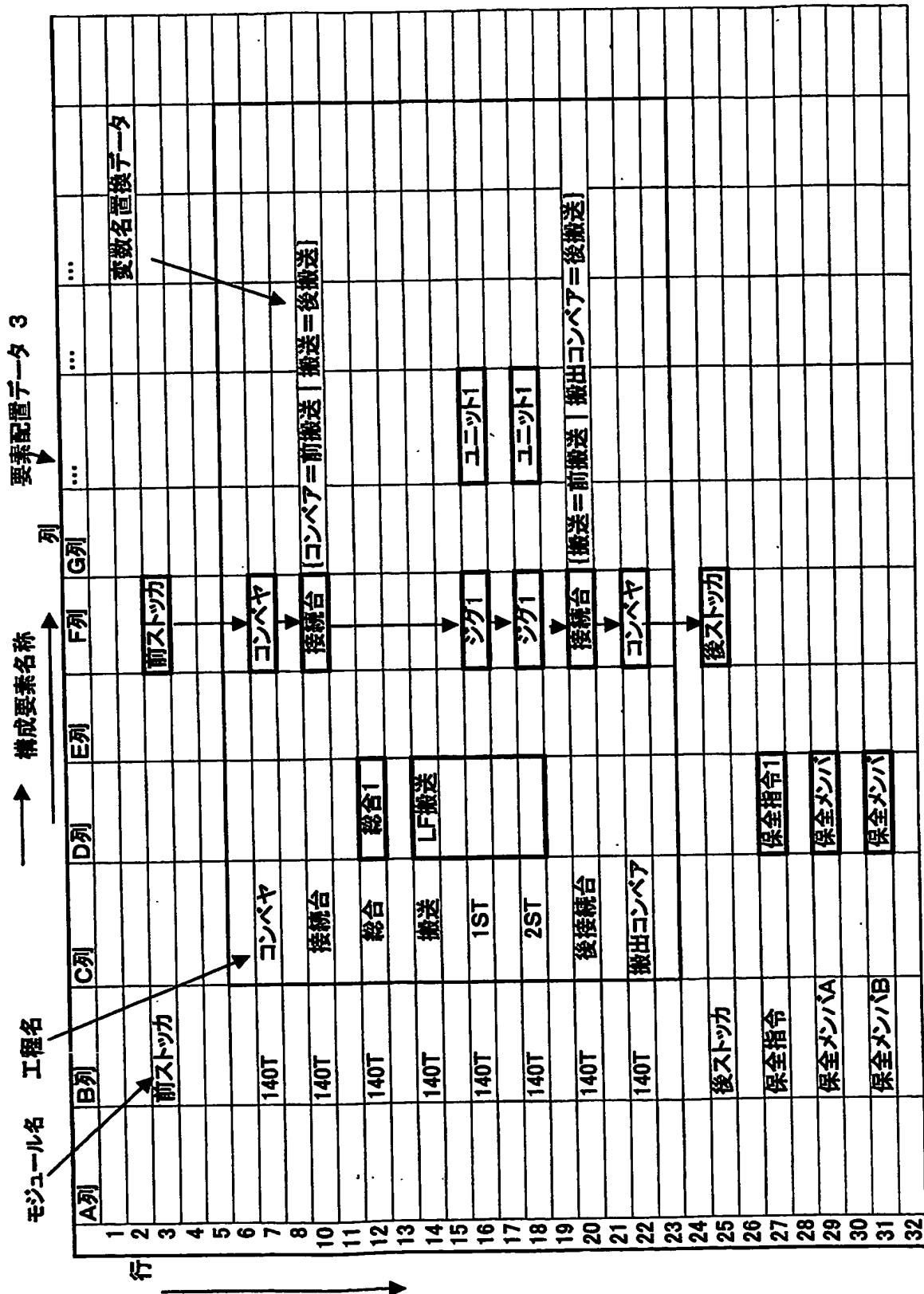
- 4 . 4 a ~ 4 j 要素定義ファイル
- 4 B 変数記述部
- 4 C プログラム記述部
- 1 0 プログラム実行手段

【書類名】 図面

【図 1】

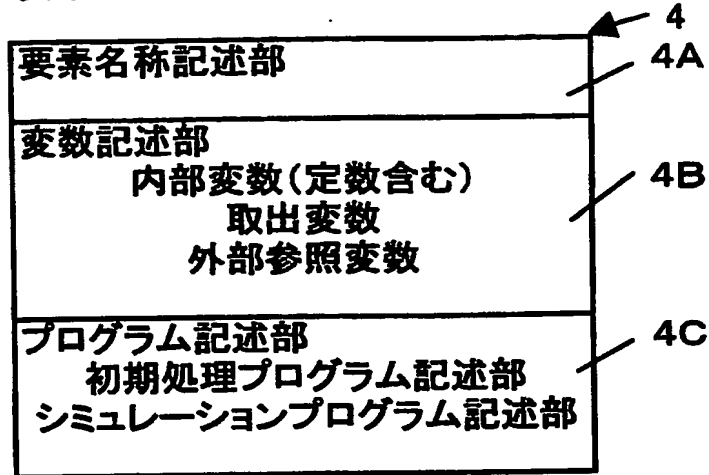


【図 2】

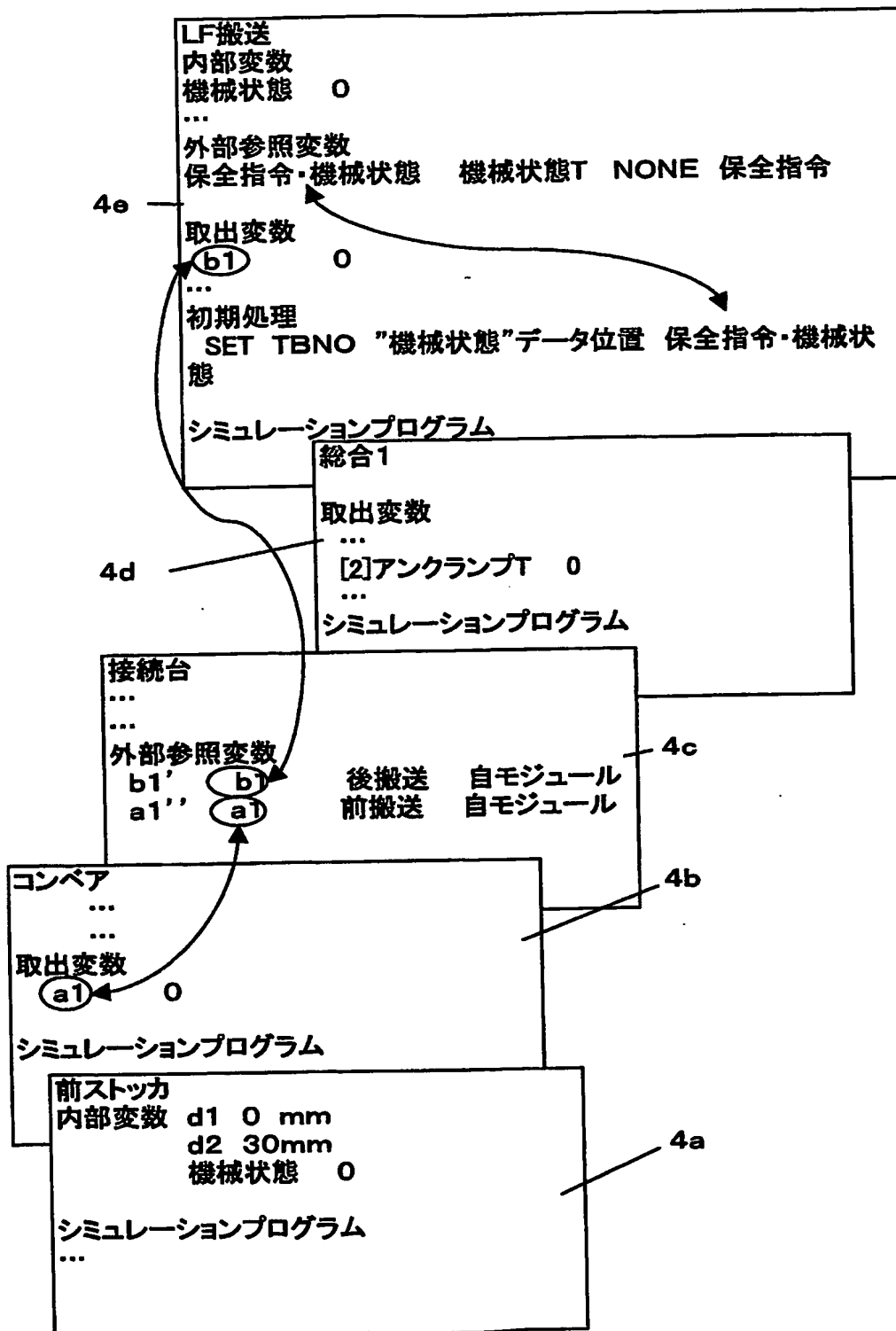


【図 3】

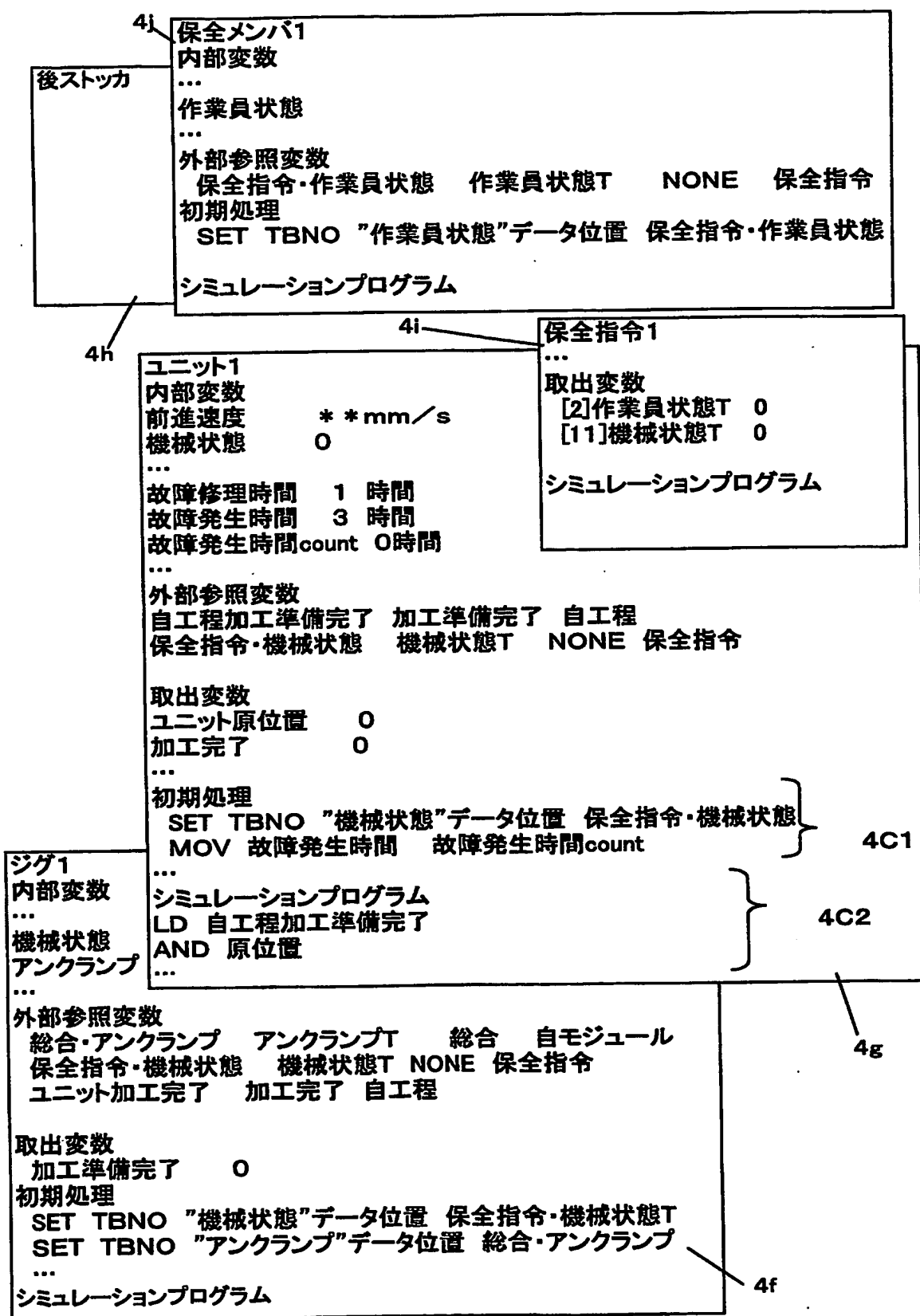
要素定義ファイルの一般的形態



【図 4】

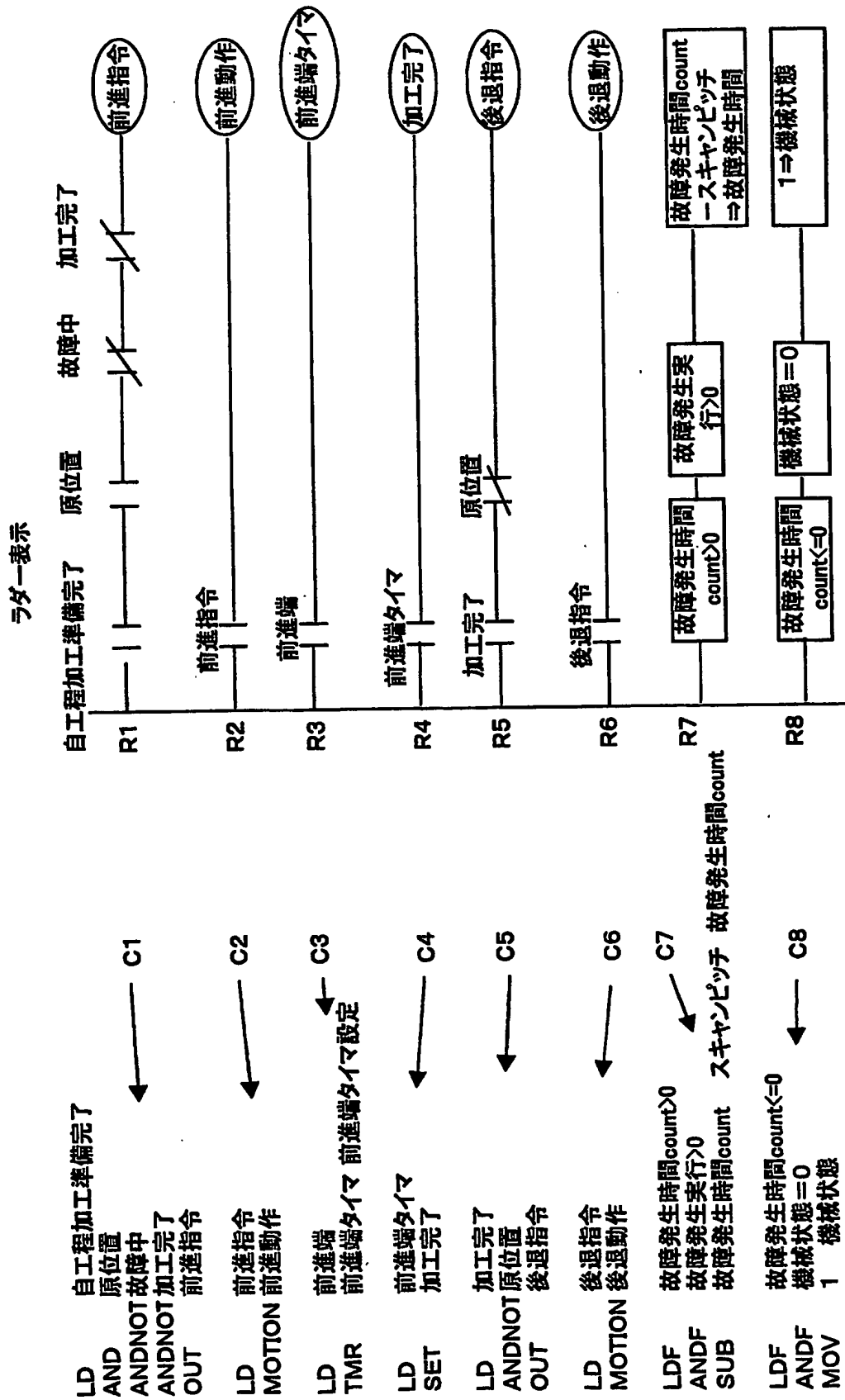


【図 5】

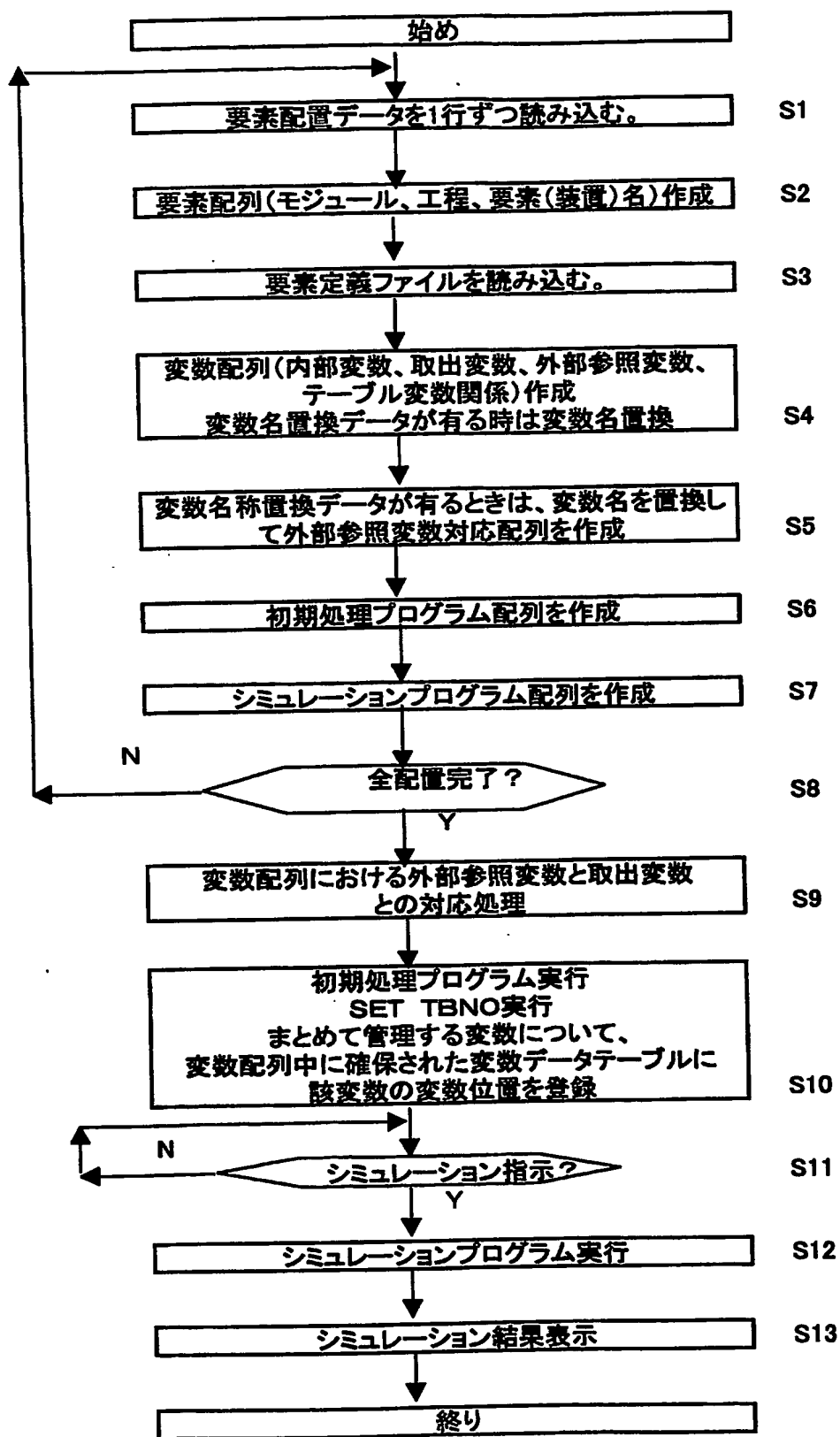


【図 6】

シミュレーションプログラムの記述



【図 7】



【図 8】

要素配列5 ↓

工程識別名称

要素配列内位置	モジュール名	工程名	要素名	変数先頭位置	個数
1	前ストック		前ストック	1	P0
2	140T	コンベア	コンベア	n1+1	P1
3	140T	接続台	接続台	n2+1	P2
4	140T	総合	総合1	n3+1	...
5	140T	搬送	LF搬送
6	140T	1ST	ジグ1
7	140T	1ST	ユニット1		
8	140T	2ST	ジグ1		
9	140T	2ST	ユニット1		
10	140T	後接続台	接続台		
11	140T	搬出コンベア	コンベア		
12	後ストック		後ストック		
13	保全指令		保全指令1		
14	保全メンバA		保全メンバ
15	保全メンバB		保全メンバ

【図 9】

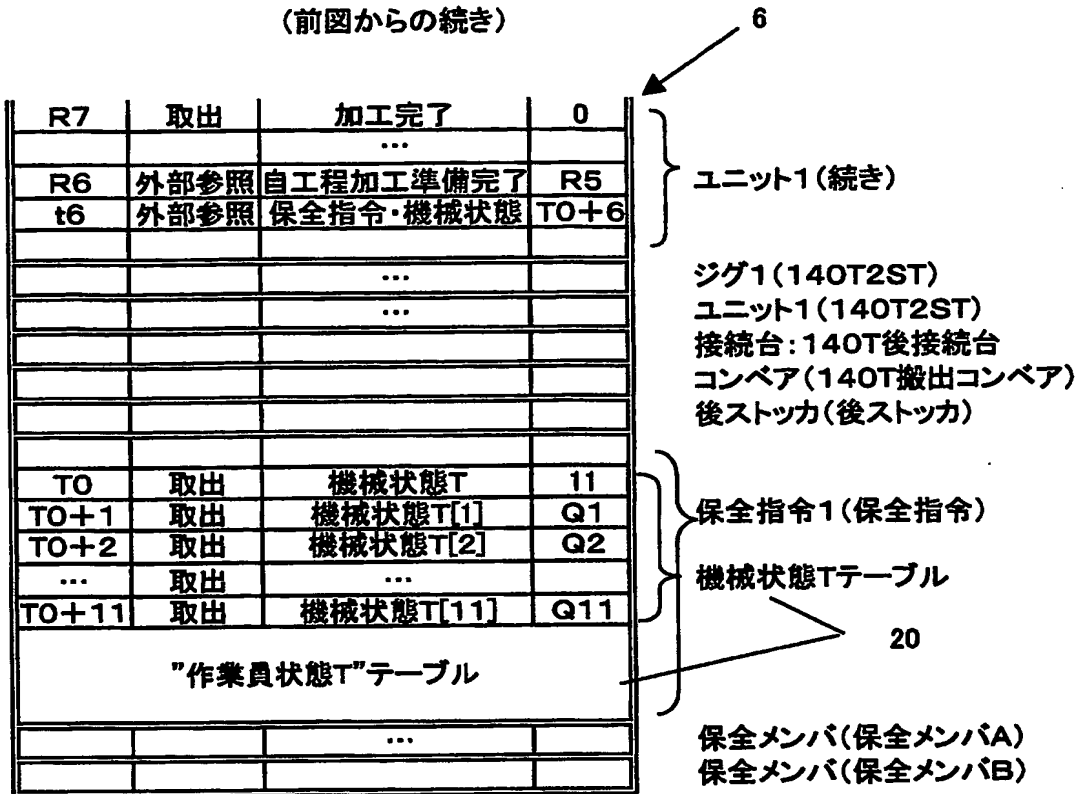
変数配列 6

変数位置 (アドレス)	属性	変数名	値/位置	
1	内部	d1	0mm	前ストッカ
2	内部	d2	30mm	
		...		
Q1	内部	機械状態	0	
		...		コンベア(140Tコンベア)
n1+1		...		
R3	取出	a1	***	
		...		接続台(140T接続台)
n2+1		...		
	外部参照	b1'	R4	
	外部参照	a1''	R3	
		...		総合1(140T総合) アंकランプテーブル 20
n3+1		...		
L0	取出	アंकランプT[0]	2	
L0+1	取出	アंकランプT[1]	f5	
L0+2	取出	アंकランプT[2]	...	LF搬送(140T搬送)
n4+1		...		
R4	取出	b1	**	
		...		
		...		ジグ1(140T1ST)
n5+1	内部	...		
Q5	内部	機械状態	0	
f5	内部	アंकランプ	0	
...	外部参照	総合・アंकランプ	L0+1	
...	外部参照	保全指令・機械状態	T0+5	
...	外部参照	ユニット加工完了	R7	
R5	取出	加工準備完了	0	ユニット1(140T1ST)
		...		
n6+1	内部	前進速度	* mm/s	
Q6	内部	機械状態	0	
		...		
	内部	故障発生時間	1時間	
	内部	故障修理時間	3時間	
	内部	故障発生時間count	0時間	
	内部	故障発生実行	1(有り)	
		...		
	取出	ユニット原位置	**	

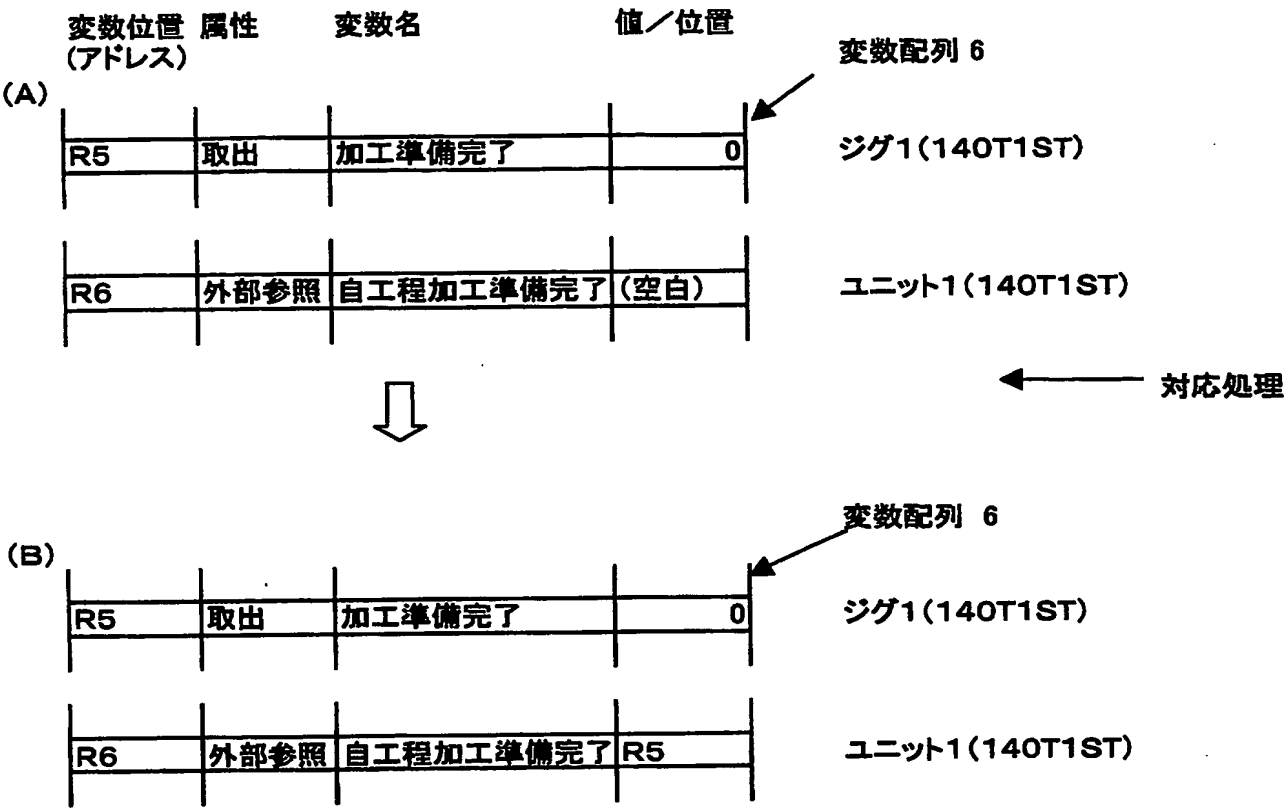
(次図へ続く)

【図 10】

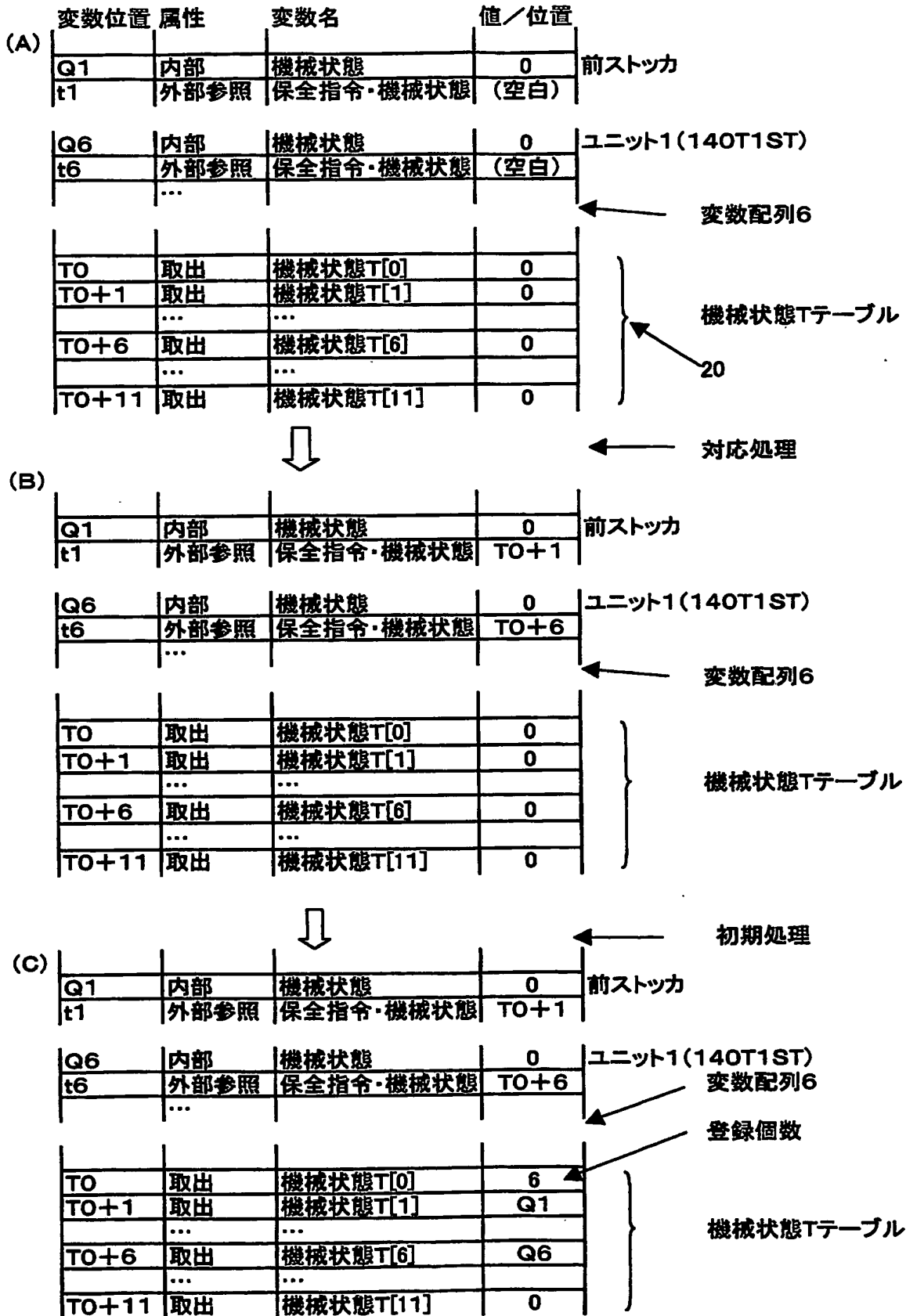
(前図からの続き)



【図 1 1】



【図 12】



【図 13】

外部参照変数対応配列 7

要素配列内位置	外部参照変数名	参照先変数名	参照先モジュール名工程名
3	b1'	b1	140T搬送
3	a1"	a1	140Tコンペア
...		...	
...		...	
7	自工程加工準備完了	加工準備完了	140T1ST
...			

【図 14】

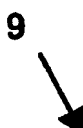
初期処理プログラム配列

8

前ストッカ
コンペア
接続台
総合1
LF搬送
ジグ1
ユニット1
ジグ1
ユニット1
接続台
コンペア
後ストッカ
保全指令
保全メンバ
保全メンバ

【図 15】

9

シミュレーション
プログラム配列

前ストッカ
コンペア
接続台
総合1
LF搬送
ジグ1
ユニット1
ジグ1
ユニット1
接続台
コンペア
後ストッカ
保全指令
保全メンバ
保全メンバ

【書類名】 要約書

【要約】

【課題】 シミュレーションプログラム生成を容易にする。

【解決手段】 工程識別名称と構成要素名称とを組み合わせる表形式の要素配置データ 3 を準備し、構成要素を定義した要素定義ファイル 4 を準備し、要素配置データ 3 を読み込みつつ、構成要素名称に対応する要素定義ファイル 4 を読み込んで、要素定義ファイル 4 に記述されている変数を変数配列 6 に並べ、初期処理プログラムを実行して互いに参照する変数間に対応付けし、同時に、シミュレーションプログラム配列 9 を生成する。プログラム実行手段 10 によりシミュレーションプログラム配列に並んだシミュレーションプログラムが実行される。

【選択図】 図 1

特願 2 0 0 3 - 3 8 0 6 5 6

出 願 人 履 歴 情 報

識別番号 [0 0 0 2 4 1 5 8 8]

1. 変更年月日 2 0 0 2 年 7 月 3 0 日

[変更理由] 住所変更

住 所 愛知県西春日井郡新川町大字須ヶ口 1 9 0 0 番地 1

氏 名 豊和工業株式会社